CHAPTER 21

Telemetry Network Standard Introduction

Acronyms **21-iii**

Chapter 21. Telemetry Network Standard Introduction **21-1**

21.1 Introduction **21-1**
21.2 Telemetry Network Standard Overview **21-2**

21.2.1 TmNS System Concepts 21-3
21.2.2 TmNS Core Technologies 21-5
21.2.3 TmNS Definitions 21-9

21.3 Relationship Between Standards and Specifications **21-15**

Appendix 21-A. Clarifications and Conventions **A-1**

A.1. Standards Key Words **A-1**
A.2. Document Conventions **A-1**

A.2.a. Usage of Defined Terms A-1
A.2.b. Usage of Message Fields A-1
A.2.c. Scope of References A-2
A.2.d. Usage of Note Boxes A-2

A.3. SNMP Conventions **A-2**
A.4. XML Concepts and Style Guide **A-3**

A.4.a. Standards Language A-3
A.4.b. General MDL Requirements A-3
A.4.c. XML Schema Concepts A-4
A.4.d. Syntax Conventions of MDL Element Descriptions A-5
A.4.e. Conditional Element in the MDL Schema Definition File A-5
A.4.f. Naming Conventions in the MDL Schema Definition File A-5
A.4.g. Indexing Policies A-5
A.4.h. Use of Supplemental XML-Based Standards A-5
A.4.i. Uniqueness of ID Attributes A-6
A.4.j. Description of ReadOnly Element A-6
A.4.k. Description of Owner Element A-6

Appendix 21-B. Bit Numbering and Byte Ordering **B-1**

B.1. Bit Field Syntax **B-1**
B.2. Bit Numbering Convention **B-1**
B.3. Octet (Byte) Ordering **B-2**

Appendix 21-C. Citations **C-1**

List of Figures

Figure 21-1. Generalized TmNS System Diagram Showing Different Control Planes 21-4
Figure 21-2. IETF Hourglass 21-5
Figure 21-3. TmNS-Specific IETF Hourglass 21-5
Figure 21-4. System Management Technologies 21-6
Figure 21-5. Core TmNS Technologies and TmNS-Specific Protocols in the TCP/IP Model Context 21-9

Acronyms

DAU	data acquisition unit
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
iNET	integrated Network Enhanced Telemetry
IP	Internet Protocol
lsb	least significant bit
LTC	Latency/Throughput Critical
MDL	Metadata Description Language
MIB	management information base
msb	most significant bit
OSI	Open Systems Interconnection
PCM	pulse code modulation
QoS	Quality of Service
RC	Reliability Critical
RF	radio frequency
RFC	Request for Comment
SNMP	Simple Network Management Protocol
TA	test article
TCP	Transmission Control Protocol
TmNS	Telemetry Network Standard
UDP	User Datagram Protocol
XML	eXtensible Markup Language

This page intentionally left blank.

CHAPTER 21

Telemetry Network Standard Introduction

21.1 Introduction

The Telemetry Network Standard (TmNS) crosses IRIG 106, chapters 21 through 28. This chapter introduces fundamental concepts and terminology used in the subsequent chapters. Additionally, this chapter provides guidance as to which of the remaining chapters might be of most interest for a particular reader. In order to guide the reader toward the chapters of further interest, the applicable chapters are referenced throughout this chapter as it introduces concepts and terminology. A quick synopsis of chapters 22 through 28 is provided below.

- IRIG 106 Chapter 22: Network-Based Protocol Suite
  The TmNS approach leverages existing standardized Internet protocols to serve as the core set of communication protocols. The TmNS's network-based protocol suite and a large portion of the Transmission Control Protocol (TCP)/Internet Protocol (IP) Protocol Suite (also known as the Internet Protocol Suite) along with other supporting technologies (e.g., underlying data link and physical layer technologies) are described in this chapter.
- IRIG 106 Chapter 23: Metadata Configuration
  This chapter describes system configuration data for TmNS-based systems. It allows them to be described in a common fashion, and it provides the means for describing the configuration of the components in a telemetry system, as well as their logical and physical interrelationships. This chapter defines a language, the Metadata Description Language (MDL), which has a syntax that defines vocabulary and sentence structure, while the MDL semantics provide meaning. The MDL provides a common exchange language that facilitates the interchange of configuration information between telemetry system components.
- IRIG 106 Chapter 24: Message Formats
  The TmNS has defined several message structures unique for its use. This chapter describes the message formats of TmNS-specific messages.
- IRIG 106 Chapter 25: Management Resources
  The TmNS defines Management Resources as resources that contain application-specific data accessible via an application layer protocol. Each TmNS application defines a set of common resources and a set of application-specific resources. This chapter provides details concerning the standardized application resources.
- IRIG 106 Chapter 26: TmNSDataMessage Transfer Protocol
  The TmNS has defined several data transfer protocols unique for its use. This chapter defines how TmNS-specific messages (TmNSDataMessages) are transferred between TmNSApps.
- IRIG 106 Chapter 27: Radio Frequency (RF) Network Access Layer
  This chapter defines the standard for managing the physical layer of RF links with the RF network. The network implements an Open Systems Interconnection (OSI) model approach to data transmission, where data moves through the OSI stack from the

application layer to the physical layer, from physical layer to physical layer through some transmission medium, then back up the stack to another application on the receiving side.

- IRIG 106 Chapter 28: RF Network Management
  This chapter defines the mechanisms and processes for managing RF links within the RF network.

21.2 Telemetry Network Standard Overview

At its core the TmNS describes networks and interfaces for components on the networks. All TmNS-based networks strive to be similar to existing Internet-based networks. Additionally, the TmNS provides mechanisms for melding with pre-existing devices, approaches, and technologies.

A fundamental principle of the TmNS approach is to enhance, rather than replace, today's telemetry systems by providing significant improvements in spectrum efficiency in order to revolutionize how flight tests are executed. This enhancement principle in turn drives the need for the new TmNS-based capabilities to be melded with pre-existing devices, approaches, and technologies. As such, the existing pulse code modulation (PCM) telemetry systems are augmented with TmNS features provided through TmNS components.

The IP network foundation of the TmNS brings with it features including routing, Quality of Service (QoS), and congestion control. The following list highlights some of the capabilities that IP networking brings to telemetry.

- Addition of Bidirectional Communications to Telemetry: bidirectional communications is one of the most fundamental enhancements provided by the TmNS. It enables the following capabilities.
  - Real-Time Access to Test Article (TA) Measurements: Provides a mechanism for real-time access to current and past measurements on the TA both directly from the sensors and from the recorder(s).
  - PCM Backfill: Provides the ability to retrieve measurements from the TAs in near real time that were dropped in the Serial Streaming Telemetry feed (PCM dropouts).
  - Real-Time Command and Control of TA Equipment: Provides the ability to status, configure, and control TA equipment from the ground station.

- Dynamic Spectrum Sharing: Provides the ability to share spectrum resources among many concurrent test activities based on instantaneous demand for telemetry resources.
- Quality of Service: Provides the ability to dynamically share spectrum resources based on priorities of certain activities over others and also to prioritize the delivery of certain measurements over others (e.g., voice).
- Fully Interconnected System: Provides the ability to seamlessly transition transmission and receipt of data from TAs from one antenna to another, including antennas in different networks (frequencies) and in other ranges. The TmNS uses the term "handoff" to describe this type of transition.

- Over-the-Horizon Telemetry: Provides the ability to perform TA-to-TA telemetry (relay) communications to support tests involving large numbers of TAs and long distances.

21.2.1 TmNS System Concepts

The TmNS defines interfaces, data delivery protocols, configuration files, and command and control concepts. These are standardized so as to support interoperability across components (and vendors) within a particular TmNS-defined network.

21.2.1.1 TmNS Interfaces

The TmNS is composed of sets of components that are modeled after network appliances typically found on the Internet. In fact, some TmNS system components (e.g., routers and switches) are almost exact functional matches to network appliances that are used on the Internet. Each TmNS-compliant component implements certain TmNS interfaces (as applicable), thus providing multi-vendor interoperability. These TmNS interfaces are as follows.

- Management Interface: Used for configuring, obtaining status, controlling, and reporting. The MDL is the main interface used for configuring TmNS-compliant devices. Further details concerning this topic are found in Chapter 23 and Chapter 25.
- Time Interface: Used for distribution and acquisition of time through the network. Further details concerning this topic are found in Chapter 22.
- Data (Measurements) Delivery Interface: Used to move acquired test data from TAs to ground processing based on different delivery requirements. Further details concerning this topic are found in Chapter 23, Chapter 24, and Chapter 26.
- RF Network Interface: Defines mechanisms for low-level control and status of the two-way telemetry communications and overall spectrum sharing. Further details concerning this topic are found in Chapter 27 and Chapter 28.

NOTE	Not all components are required to support all interfaces. For example, a data acquisition unit (DAU) would typically implement the management, time, and data interfaces listed above. This architecture choice was made to minimize the complexity of any one item and to aid the possibility of creating a broad array of configurations.

21.2.1.2 Data Delivery

The TmNS defines two data delivery mechanisms.

- Latency/Throughput Critical Delivery Protocol: used to deliver test data when latency or throughput constraints are more important than reliability constraints (real-time). The underlying technologies supporting this delivery protocol are User Datagram Protocol, Internet Group Management Protocol, and IP multicasting.
- Reliability Critical Delivery Protocol: used to deliver test data when reliability constraints are more important than latency or throughput constraints (reliable). The underlying technologies supporting this delivery protocol are TCP and Real Time Streaming Protocol. Further details concerning this topic are found in Chapter 26.

NOTE	Data delivery concepts support variations for latency, throughput, and reliability. For instance, during one phase of a particular test, the test operators may need samples of a particular set of measurements with as little latency as possible due to safety of flight issues, even if it means losing some samples during telemetry dropouts. In another phase of the same test, the test operators may need a reliable transport of these same measurements for analysis even if it raises latency due to resending data lost during telemetry dropouts.

21.2.1.3 Command and Control Planes

The TmNS defines two primary command and control planes.

- Test/Mission Command and Control Plane (Red Network): This plane is focused on command and control associated with a particular test. It is concerned with measurements, telemetry processing, message/data formats, data recording, and TA component configuration and status. This plane resides in the red-side (plain-text) portions of a TmNS system, which are mainly comprised of the red network components on the TA(s) and Mission Control Room, as shown in Figure 21-1. Red Network components are behind a Type-1 inline network encryptor.

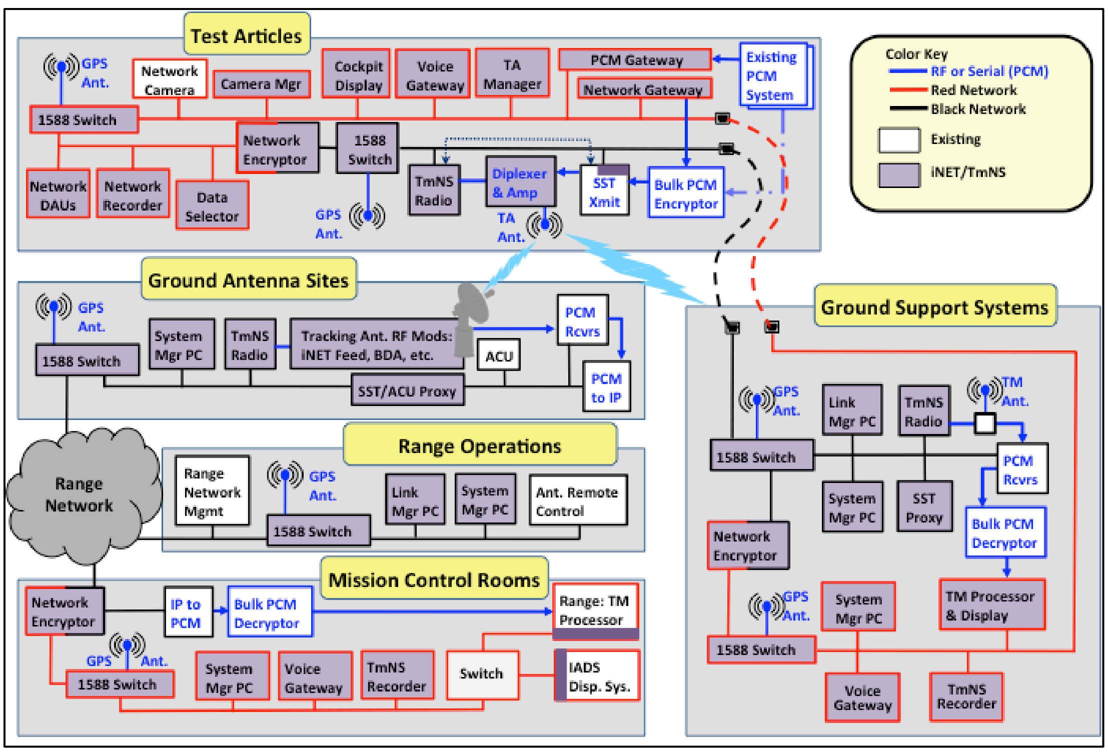

Figure 21-1. Generalized TmNS System Diagram Showing Different Control Planes

- Range Infrastructure Command and Control Plane (Black Network): This plane is focused on command and control associated with the provisioning of resources needed for a given test or set of tests within a range or across multiple ranges. It is

concerned with spectrum sharing, QoS, establishment and management of two-way telemetry communications, and the transitioning of communications from TAs from a given ground antenna site to another (antenna-to-antenna handoff). This plane resides in the black-side (cypher-text) portions of a TmNS system, which are mainly comprised of the ground antenna sites, range operations center, and black network components on the TA(s), as shown in Figure 21-1. Further details concerning this topic are found in Chapter 25 and Chapter 28.

NOTE	By separating the control into two planes, areas of concern may be separate across range personnel.

21.2.2 TmNS Core Technologies

The TmNS utilizes an IP network that is based on the success and description of the Internet Engineering Task Force (IETF) hourglass approach, as shown in Figure 21-2. The IP layer is the basic interoperability between networked components. Figure 21-3 shows a TmNS specialization of the classic IETF IP hourglass figure.

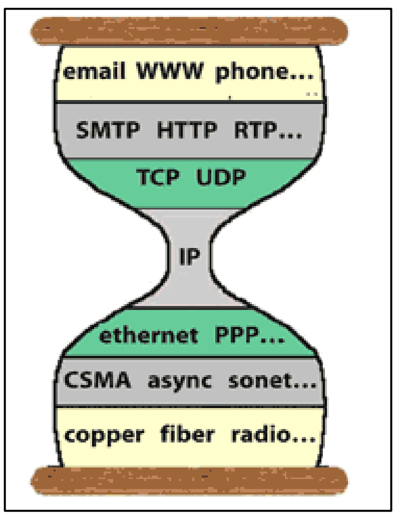

Figure 21-2. IETF Hourglass

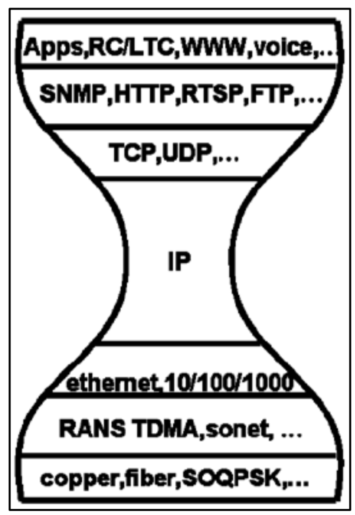

Figure 21-3. TmNS-Specific IETF Hourglass

Further details concerning this topic are found in Chapter 22.

21.2.2.1 Network-Based Data Messages

Test data is delivered in TmNSDataMessages, which contain a header and a payload. Actual measurements are contained in the packages within a TmNSDataMessage, and the mapping of measurements in a TmNSDataMessage is defined in a system configuration file, which is an MDL file that describes the configuration for a particular device that is transmitting or consuming a given TmNSDataMessage. Further details concerning this topic are found in Chapter 23 and Chapter 24.

21.2.2.2 System Configuration and Management

System management within the TmNS is based upon the International Organization for Standardization Telecommunications Management Network model FCAPS, which stands for fault, configuration, accounting, performance, and security.

System management is used across a TmNS system to manage TmNS-compliant components, providing a view of fault, configuration, accounting, performance, and security configuration information on the network. Essentially, a TmNS system is composed of two types of components when it comes to management and configuration:

1. Managed devices: Any TmNS-compliant component that provides a management interface as defined by Chapter 25;
2. TmNS Managers: An entity that manages TmNS-compliant components. Managers implement the interfaces necessary to manage TmNS-compliant components in accordance with Chapter 25. Further details concerning this topic are found in Chapter 25.

Figure 21-4 shows the building blocks of system management as specified by the TmNS. The core technologies used are Simple Network Management Protocol (SNMP) to pass management information through the system. The SNMP management information bases (MIBs) provide dictionaries for management information. Managed devices execute applications called agents that use the TmNS-defined MIB to provide their internal status and to accept controls and configuration. File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), and Internet Control Message Protocol (ping) play supporting roles related to file transfer, discovery, and configuration.

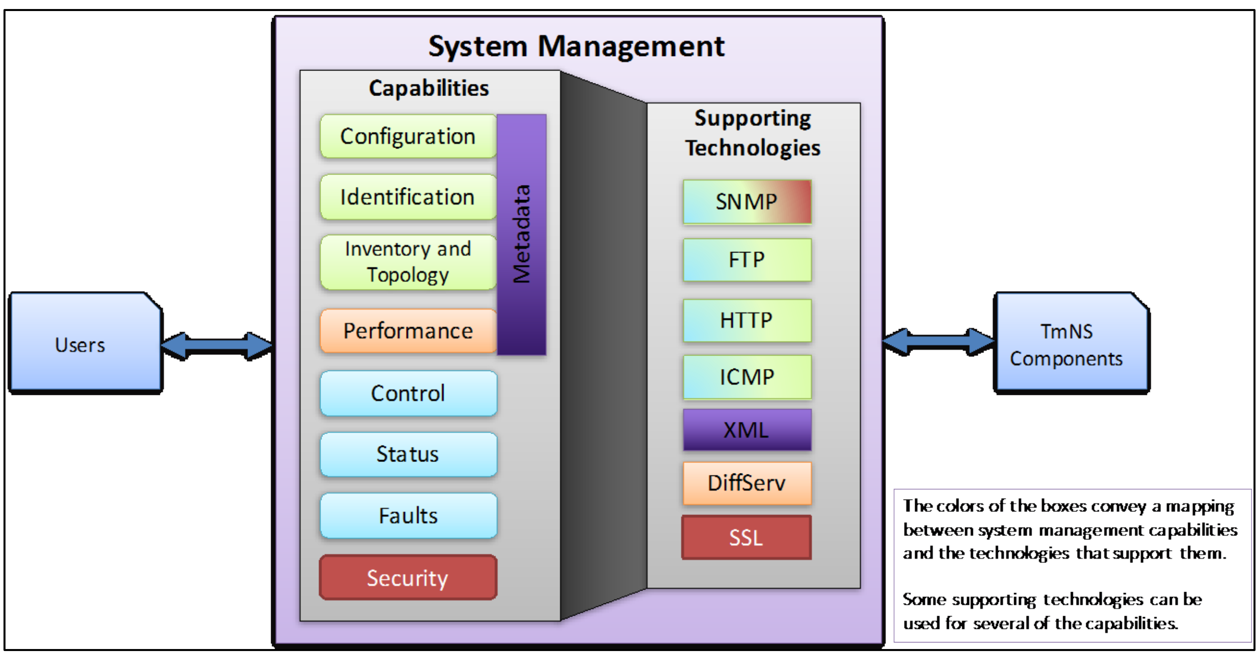

Figure 21-4. System Management Technologies

Further details concerning this topic are found in Chapter 22.

The MDL is used for describing system configuration (Metadata) in a common fashion. The eXtensible Markup Language (XML) schema defined by the TmNS provides the means for describing the configuration of the components in a TmNS system as well as their logical and physical interrelationships. The MDL is expressive enough to describe a wide variety of systems: large and small, simple and complex, from the low-level transducer-to-measurement association for an acquisition card on a DAU up to network topology of multiple test mission networks.

A table containing a mapping of MDL elements to relevant paragraphs of the TmNS (IRIG 106 Chapters 21-28) is contained in Chapter 23 Appendix 23-B. This table can be used by a reader of the standard to identify the MDL elements that correspond to particular paragraphs or to identify the paragraphs that correspond to particular MDL elements.

Further details concerning this topic are found in Chapter 23.

By using the system management capabilities, TmNS-compliant components can be configured, reconfigured, controlled, and statused in an interoperable way.

NOTE	A typical way to utilize the system management capabilities is to provide a system manager. This kind of user application provides monitoring, controlling, configuring, coordinating, and visualizing the operations of a system built based on the TmNS. System manager users are typically able to obtain system and device-level status, including status of TA instrumentation and information about local and system-wide network performance (expected versus actual). Additionally, the display of a system manager typically provides an indication of system health and details of any fault conditions detected within the TmNS system.

21.2.2.3 Time

Time within an entire TmNS-based system is distributed using IEEE 1588-2008[1], also known as Precision Time Protocol Version 2. Time within a TmNS system is delivered without the addition of any wires.

Further details concerning this topic are found in Chapter 22.

NOTE	All TmNS-compliant network switches can be synchronized to an external time source (e.g., Global Positioning System) and act as 1588 master clocks for a local network within the TmNS network (e.g., red TA network, black TA network, etc.). Components requiring sub-microsecond precision, such as DAUs for time stamping measurements, are able to do so using a hardware implementation of 1588.

[1] Institute of Electrical and Electronics Engineers. *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE 1588-2008. Geneva: International Electrotechnical Commission, 2008.

21.2.2.4 Quality of Service

The TmNS annotates a typical Differentiated Services architecture, which is a standard IP QoS mechanism for coordination of the delivery of competing data and command and control network traffic.

Further details concerning this topic are found in Chapter 22 and Chapter 23.

> **NOTE** The QoS mechanism can be used to for certain sets of data within a particular test (or across multiple tests) that might have stringent delivery requirements due to performance reasons (e.g., voice data), safety of flight concerns, etc.

21.2.2.5 Routing

Routing is the process of selecting best paths in a network. The TmNS annotates IETF standards concerning a typical routed IP network. Using the classic routed IP architecture enables a variety of advanced capabilities, including relay, and other capabilities that have not yet been explored.

Further details concerning this topic are found in Chapter 22 and Chapter 23.

> **NOTE** Just as in large-scale networks (e.g., the Internet) the components within a TmNS-based network are not aware about the network path that is used to deliver data from one node to another. All a given component needs to know is its next hop. This means that components that transport data within the TmNS system need to support these classic routing concepts, including TmNS-compliant radios, which are network routers themselves. As such, radios in general can route data to any other radio within reach at any time.

21.2.2.6 Source Selection

When RF propagation from one TmNS-compliant transmitting radio source arrives at two or more TmNS-compliant receiving radios, it is possible using routing and source selection to choose any one of the network packets. This support is provided through TmNS interfaces, data message formats, and management concepts. Collectively, these concepts are called TmNS Source Selector.

Further details concerning this topic are found in Chapter 28.

21.2.2.7 Security

The TmNS is architected with a variety of security mechanisms in order to meet a particular program's needs. The TmNS security mechanisms are segmented into mechanisms that secure the data transfer from the TAs to the ground (i.e., from one secure enclave to another), as well as for securing other types of communications where the information is not classified, but can be sensitive from an operational perspective.

Communications between secure enclaves (e.g., TAs and mission control) are protected via National Security Agency-approved type-1 security mechanisms that mitigate the anticipated threats. The RF communications are protected via FIPS-140-2 mechanisms.

Additional security mechanisms used to protect data within a TmNS system include:

- Secure Sockets Layer (SSL): used as a security mechanism for transferring data over HTTP and FTP.
- SNMP v3: needed for secure SNMP communications within a TmNS system. It supports both authentication and privacy.

Further details concerning this topic are found in Chapter 22.

21.2.2.8 Layered Architecture and Summary of Core Technologies

The TmNS architecture is, by design, a communications and data delivery system that is partitioned into abstraction layers. As in the OSI model, a layer serves the layer above it and is served by the layer below it. The layers are in general independent, so that a layer can be changed with little to no impact to the other layers. This layered architecture in turn allows different technologies to be used in each layer.

Figure 21-2 and Figure 21-3 show the IETF hourglass approach and the corresponding specialization of that hourglass. Figure 21-5 depicts the technologies discussed in this section and how they relate to each other and work cohesively across the different TCP/IP model layers. Further details concerning this topic are found in Chapter 22.

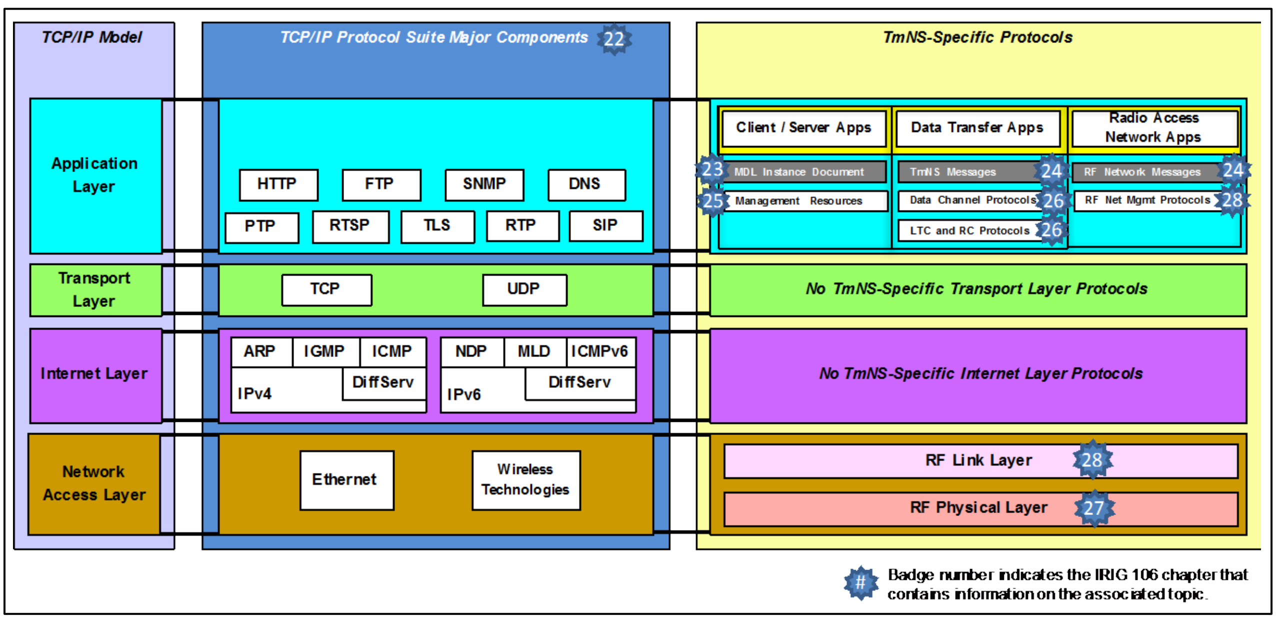

Figure 21-5. Core TmNS Technologies and TmNS-Specific Protocols in the TCP/IP Model Context

21.2.3 TmNS Definitions

The TmNS utilizes a collection of terms that have specific meanings when used in a TmNS context. They are typically highlighted in *italics*. A list of the overarching definitions appears in this section.

AES Cryptographic Algorithm: This Advanced Encryption Standard (AES) block cipher encryption algorithm, described in detail in FIPS PUB 197[2], is recommended by the National Security Agency in order to provide an adequate protection mechanism for the communication link.

Agent: A Simple Network Management Protocol (SNMP) process that provides SNMP-based *ManagementResources* on a *NetworkNode* or *NetworkDevice*.

Attached Synchronization Marker (ASM): A specific bit pattern preceding each low-density parity-check *codeblock* group to aid *codeblock frame* synchronization.

Bit Error Rate: The ratio of the number of bits incorrectly received to the total number of bits sent during a specific time interval.

Black (or Blackside): A portion of a network that is not physically protected (not secure) with respect to another portion of the network. Sensitive data that traverses this network must be protected by encryption.

Burst: The time interval of RF emission, from start to end in a time-division multiplex media access scheme.

Burst Preamble: A specific bit pattern transmitted at the beginning of a *burst* to facilitate carrier frequency symbol timing acquisition.

Burst Sequence: The *burst* information field structure.

Burst Synchronization: Involves the acquisition and tracking of the signal carrier(s), the symbols/bits, the frames or *codeblocks* from a recovered clock at the receiver.

Carrier Frequency Error: Uplink and downlink frequency error bounds established for single-carrier *SOQPSK-TG waveform*.

Codeblock: The minimum quanta of a fixed LDPC codeword block that consists of 4,096 information bits or 6144 coded bits with 2/3 LDPC code rate.

Codeblock Frame: A variable PHY frame unit that consists of a minimum of one LDPC *codeblock* and up to maximum of eight LDPC *codeblocks*. It is preceded by an *attached synchronization marker* (ASM).

DataDeliveryControlChannel: The common elements of the communication mechanisms for the setup, tear-down, and operation of the *RC* and *LTC Delivery Protocols*. See Chapter 26.

DataChannel: Identifies a network connection used to transport *TmNSDataMessages* between a *DataSource* and a *DataSink.*

DataSink: A *TmNSApp* that consumes *TmNSDataMessages* that contain *MeasurementData*. Identified as the data-consuming portion of a *ResourceClient* or *ResourceServer*.

DataSource: A *TmNSApp* that produces *TmNSDataMessages* that contain *MeasurementData*. Identified as the data-producing portion of a *ResourceClient* or *ResourceServer*.

[2] National Institute of Standards and Technology. "Specification for the Advanced Encryption Standard (AES)." FIPS PUB 197. 26 November, 2001. May be superseded by update. Available at http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

DiffServ (Differentiated Services): A computer networking architecture that specifies a simple, scalable and coarse-grained mechanism for classifying, managing and providing *Quality of Service* (QoS) guarantees on IP network traffic.

Downlink Transmission: Communication originating at a *Test Article* and terminating at the *Ground*. With reference to a *Relay*, communication originating at a *Test Article* and terminating at the *Relay*.

Dynamic Allocation: A method of scheduling TDMA time slots for transmissions by radios based on a set of criteria, such as bandwidth needs and mission priorities.

Enclave: A distinct portion of a network, system or facility that is isolated, usually for security-related purposes, from the rest of the network, system or facility.

Encryption & Decryption: NIST FIPS 140-2 certified bulk cryptographic module along with *AES cryptographic algorithm* is recommended by the National Security Agency for communication link security at link layer.

Epoch: A TDMA frame unit that allocates *transmission opportunity* (TxOp) resources for uplink and downlink. Epoch is equivalent to a *TDMA frame*.

Forward Error Correction: A system of error control for data transmission, whereby sender adds redundant data to its messages, which is known as error correction code. This allows receiver to detect and correct errors (within some bound) without the need to ask the sender for additional data.

Ground Network: One or more *TmNS Networks* that interconnect *Ground*-based *NetworkNodes*.

Ground Station (GS): A ground infrastructure that, at a minimum, consists of primary and remote antenna sites, serial streaming telemetry (SST) and *ground network* infrastructures. Nominally *Ground Radios* are located in a *Ground Station*.

Ground Station Network: A *TmNS Network* that interconnects connected *NetworkNodes* physically residing in a *Ground Station*.

Handoff: The process of transferring communications from one source radio to another source radio for the same destination RF MAC Address. The original source radio may be referred to as the “Leave Radio” while the new source radio may be referred to as the “Join Radio”.

HDLC Frame: A protocol based on ISO-13239 Standard that was modified to support frame boundary delineations, to carry link layer control messages and user datagrams.

Information Data: The channel information data, prior to channel coding, that includes user data and channel overhead affiliated with OSI layer-1 and layer-2. Overhead affiliated with OSI layer-3 through layer-7 is included as user data.

Integrated Services (IntServ): A computer network architecture that specifies fine-grained, reservation-based mechanisms for providing *Quality of Service* (QoS) guarantees for individual IP network traffic flows.

Latency/Throughput Critical (LTC) Delivery Protocol: The TmNS-specific application-level method of delivering *TmNSDataMessages* via User Datagram Protocol (UDP).

Link Agent: Executes link control operations in a *Radio*.

Link Manager (LM): A *TmNSApp* responsible for optimized control and coordination of *Radio* operations across multiple *Radios* in the RF Network. The primary role of RF Link Management is implementation of the TDMA controller that allocates transmission opportunities for its managed *Radio* components.

Low-Density Parity-Check (LDPC) Code – A variant of *Forward Error Correction* codes that uses block codes for error correction. Code is specified by parity check matrix *H* and utilizes generator matrix *G* to perform encoding.

LTCControlChannel: The communication mechanism for the setup, tear-down, and operation of the *LTC Delivery Protocol*. See Chapter 26.

LTCDataChannel: The communication mechanism for delivery of *TmNSDataMessages* using the *LTC Delivery Protocol*. See Chapter 26.

LTCDataSink: A *DataSink* that utilizes the *LTC Delivery Protocol*.

LTCDataSource: A *DataSource* that utilizes the *LTC Delivery Protocol*.

Management Information Base (MIB): A “Structure of Management Information” (SMI) formatted text file used by the SNMP *Agents* and *Managers* to define a common communication language for exchanging management information.

ManagementResource: An application-accessible entity that is used for command, control, and health and status monitoring. *ManagementResources* may be generic to the host platform or may be specific to the TmNS-based environment.

ManagementURI: The Uniform Resource Identifier (URI) that describes a particular *ManagementResource*.

Manager: A Simple Network Management Protocol (SNMP) process that accesses SNMP-based *ManagementResources* on a *NetworkNode*.

MeasurementData: A digital representation of a measurement.

MeasurementID (MeasID): A numerical identifier that refers to a specific *MeasurementData* described in an MDL instance document.

MessageDefinitionID (MDID): A numerical identifier that refers to a specific *Message* Definition described in an MDL instance document.

Metadata: Information that describes a system and data interrelationships; defined in the Telemetry Network Standard.

Metadata Description Language (MDL) Instance Document: A document that complies with the language defined in Chapter 23.

NetworkDevice: A *NetworkNode* that provides network and/or data link layer service and interconnectivity, without modifying data above the network layer. See Open Systems Interconnection (OSI) model.

NetworkInterface: A module that implements an interface, both logical and physical, between a *NetworkNode* and a *TmNS Network*.

NetworkNode: Any device that contains a *NetworkInterface* that is connected to a *TmNS Network*. Nominally runs one or more *TmNSApps*.

Notification: An asynchronous SNMP message generated by a *TMA*.

Occupied Bandwidth (OBW): The bandwidth containing 99% of the total integrated power of the transmitted spectrum, centered on the assigned channel frequency. The width of a frequency band such that, below the lower and above the upper frequency limits, the mean powers emitted are each equal to a specified percentage B/2 of the total mean power of a given emission. In this standard, B/2 is taken as 0.5%.

Octet: A sequence of eight bits.

Package: A data container composed of *MeasurementData*.

PackageDefinitionID (PDID): A numerical identifier that refers to a specific *Package* Definition described in an MDL instance document.

Physical Layer (PHY): The first and lowest layer in the seven-layer OSI model. This layer defines the means of transmitting raw bits rather than logical data packets over a physical link connecting Radio and network nodes. This layer translates logical communications requests from the data link layer into hardware-specific operations to effect transmission or reception of electronic signals.

Quality of Service: An umbrella term describing the delivery and performance requirements of a data transfer and/or the network mechanisms used to meet those requirements.

Queue Management: An RF Network-defined common, standardized interface to the *Traffic Engineering Queues*, which may be implemented with non-standard, vendor-specific mechanisms.

Radio: Consists of a *Link Agent*, RF transceiver, and Ethernet transceiver.

Radio Air Channel Data Rate: Raw channel data rate that includes user data, aggregated overheads (physical and layer-2), and coding overhead.

Radio Air Data Rate: Data rate from the output of the radio modulator. Specified as:

- Radio air user data rate, prior to aggregated overheads (physical and layer-2) and coding.
- Radio air information data rate that includes aggregated overheads but prior to coding.
- Radio air channel data rate that includes aggregated overheads and coding

Radio Bearer: The service provided by the RF Network to transfer data between the test article network and ground station network. Service is the collection of all means and facilities provided by the network to allow a certain type of communication over the network.

RCControlChannel: The communication mechanism for the setup, tear-down, and operation of the *RC Delivery Protocol*. See Chapter 26.

RCDataChannel: The communication mechanism for delivery of *TmNSDataMessages* using the *RC Delivery Protocol*. See Chapter 26.

RCDataSink: A *DataSink* that utilizes the *RC Delivery Protocol*.

RCDataSource: A *DataSource* that utilizes the *RC Delivery Protocol*.

Red (or Redside): A portion of a network that is physically protected (secure) with respect to another portion of the network. Sensitive data may be communicated within this protected enclave without need for encryption.

Relay: Hierarchical TDMA node structure that allows *Test Article* to act as a relay node to extend communication link ranges by facilitating nearby *Test Articles* to join the network and by linking communications between *TDMA controllers*.

Reliability Critical (RC) Delivery Protocol: The TmNS-specific application-level method of delivering *TmNSDataMessages* via Transmission Control Protocol (TCP).

ResourceChannel: Identifies a network connection used to transport *ResourceRequests* and *ResourceResponses* between a *ResourceClient* and a *ResourceServer*.

ResourceClient: A *TmNSApp* that generates *ResourceRequests* and may incorporate the *DataSource* and/or *DataSink* functionality.

ResourceInterface: A software interface used by *TMAs* to access *ManagementResources*. The standard currently supports an SNMP-based interface and an HTTP-based interface for accessing different *ManagementResources*.

ResourceRequest: Request to access a specific *ManagementResource* and is generated by a *ResourceClient*.

ResourceResponse: Returns information in response to a *ResourceRequest* regarding a specific *ManagementResource* and is generated by a *ResourceServer*.

ResourceServer: A *TMA* that receives and processes *ResourceRequests*, generates *ResourceResponses*, and may incorporate the *DataSource* and/or *DataSink* functionality.

RF Network: The segment of a *TmNS Network* that provides network connectivity over RF interfaces between *Test Article Networks* and *Ground Station Networks*.

RF Network Message (RFNM): A network-independent structure that contains control or status information regarding RF Network conditions.

RoleID: A string that refers to the role of a *TMA*.

SOQPSK-TG Waveform: An RCC-TG-defined variant of MIL-STD-188/181A ternary continuous phase modulated single-carrier waveform established to achieve spectrum efficiency and robustness.

Spectral Mask: Requirement for RF emission spectrum containment for single-carrier *SOQPSK-TG waveform*.

Telemetry Network Standard (TmNS): Another name for IRIG 106 Chapters 21-28.

Test Article: A vehicle infrastructure that, at a minimum, consists of on-board antenna, Serial Streaming Telemetry (SST) and test article network infrastructures.

Test Article Network: A *TmNS Network* that interconnects connected *NetworkNodes* physically residing on a test article.

Time Division Multiple Access (TDMA): A Time-Division Duplex scheme (TDD) to separate uplink and downlink transmission signals. TDMA emulates full-duplex communication over a half-duplex link.

TmNSApplication (TmNSApp): an application running on a *NetworkNode* that provides or utilizes one or more TmNS interfaces.

TmNSManageableApplication (TMA): A TmNSApp that provides other applications with access to a set of *ManagementResources* via one or more *ResourceInterfaces*.

TmNSAppManager: An application that monitors the status or controls one or more *TMAs*.

TmNSDataMessage: An MDID-based *TmNSMessage* that contains a *TmNSDataMessageHeader* and a *TmNSDataMessagePayload*; described in Chapter 24.

TmNSDataMessageHeader: Fields in a *TmNSDataMessage* that precede a *TmNSDataMessagePayload*.

TmNSDataMessagePayload: Composed of zero or more *Packages*.

TmNSMessage: A network-independent structure composed of a *TmNSMessageHeader* and a *TmNSMessagePayload*; described in Chapter 24.

TmNStimestamp: A TmNS-specific time format for encoding timestamps in a human-readable textual representation (yyyymmddThhmmss.sssssssss).

TmNS Network: A network that conforms to the IRIG 106 Chapter 21-28 Telemetry Network Standard.

TmNS Source Selector (TSS): Tunnel management functionality

TmNS_Request_Defined_URI: The uniform resource identifier (URI) that describes the request specification as defined by the *LTC* and *RC Delivery Protocols*.

Traffic Engineering Queues (TE Queues): A set of functionality provided by the RF Network that collectively includes the implementation and control of queue structures and associated mechanisms used to provide optimized *Quality of Service* performance.

Transmission Opportunity (TxOp): Transmission time slots assigned by a *TDMA controller* to each *Test Article Radio* for downlink transmission of data and control information and to the *Ground Station Radio* for uplink transmission of data and control information.

TSS Client: An application that implements one or more TSS Interfaces and issues tunnel connection commands to a TSS Server.

TSS Server: An application that implements a TSS Interface and listens for incoming tunnel connection commands from TSS Clients.

Type Length Value (TLV): A flexible format for defining or specifying data fields in a message, especially when the fields may be of variable length and multiple fields are encapsulated into the message. Used as the data structure that forms RFNMs.

Uplink Transmission: Communication originating at the *Ground* and terminating at a *Test Article*. With reference to a *Relay*, communication originating at the *Relay* and terminating at a *Test Article*.

User Data: Referred to as test data, mission data, or data plane data.

21.3 Relationship Between Standards and Specifications

As part of the integrated Network Enhanced Telemetry (iNET) program, the TmNS and specifications were developed to guide the development of the system and the interoperability between the components. The goal of the TmNS is to promote an open system architecture and interoperability across component vendors by defining functional system interfaces. The intent of the specifications is to define the system, hardware, software, testing, and performance requirements associated with the TmNS Demonstration System and each of the components within the TmNS Demonstration System. As such, the requirements contained in each component specification largely reference back to the TmNS. It is important to note that the specifications were developed in preparation for the TmNS Demonstration System and, while they are suited for other systems implementing the TmNS, a range may decide to tailor these specifications to meet their specific needs.

APPENDIX 21-A

Clarifications and Conventions

A.1. Standards Key Words

In many sections of Chapters 21-28, key words are used to signify the requirements in the standard. This section defines these words (derived from Request for Comment [RFC] 2119[3]) as they should be interpreted in iNET standards. Note that the force of these words is modified by the requirement level of the standard in which they are used.

- SHALL: This word means that the definition is an absolute requirement of the standard.
- SHALL NOT: This phrase means that the definition is an absolute prohibition of the standard.
- SHOULD: This word means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word means that an item is truly optional. One implementation may choose to include the item because a particular marketplace requires it or because the implementation enhances the product while another implementation may omit the same item. An implementation that does not include a particular option SHALL be prepared to interoperate with another implementation that does include the option, though perhaps with reduced functionality. In the same vein, an implementation that does include a particular option SHALL be prepared to interoperate with another implementation that does not include the option (except, of course, for the feature the option provides).

A.2. Document Conventions

A.2.a. Usage of Defined Terms

The words defined in Subsection 21.2.3 are reserved for specific use and will be italicized when they appear throughout the TmNS chapters. The use of italics is reserved exclusively for words that appear in Subsection 21.2.3.

A.2.b. Usage of Message Fields

Names of specific fields within the *TmNSDataMessage* structure are indicated by an Arial font. Some field names are the same as terms defined in Subsection 21.2.3. When a statement refers to a field, the field name will adhere to this convention. It will not be italicized.

[3] Internet Engineering Task Force. "Key Words for Use in RFCs to Indicate Requirement Level." RFC 2119. March 1997. May be superseded or amended by update. Retrieved 18 April 2017. Available at https://datatracker.ietf.org/doc/rfc2119/.

A.2.c. Scope of References

A reference to a section number from any of the TmNS chapters includes only that specific section and not its subsections. A reference to a section number followed by an asterisk indicates that the section referenced and all of its subsections are included in the context of the reference.

A.2.d. Usage of Note Boxes

Throughout the TmNS chapters, note boxes such as the one below will appear with information relevant to the material being presented in the surrounding text. These note boxes will act as a supplement to guide the reader with rationale and advisories where they are deemed useful; however, the content of the note boxes is purely informational. Either by their presence and/or removal, the note boxes shall not augment the rules and specifications presented in the TmNS in any way.

	This is an example of a note box that appears in the TmNS.

A.3. SNMP Conventions

This document uses a set of conventions when defining SNMP variables.

- For each variable, a "Type" and a "Read-Write" value is indicated. These values are defined by the SNMP RFCs and are only restated here for clarity.
  - Type (of SNMP variables) – NOTIFICATION-TYPE, IpAddress, Counter64, Counter32, Integer32, Unsigned32, and TimeTicks are defined by SNMPv2-SMI (RFC 2578[4]). TestAndIncr, TruthValue, and DisplayString are defined by SNMPv2-TC (RFC 2579[5]). INTEGER is an enumerated form of Integer32.
  - Read-Write (of SNMP variable) – read-only, read-write, read-create, not-accessible, and accessible-for-notify are SNMP variable access levels (RFC 2578). The first two types are self-explanatory. The term "read-create" indicates a table entry may be read, created, or modified. The term "not-accessible" means the variable is used internally by the SNMP *Agent* (such as a table index), but is not retrievable through SNMP network commands. The term "accessible-for-notify" means the variable is used as part of an SNMP notification and is not retrievable through SNMP network commands.

- To define the structure of the SNMP *MIB* tree, the following convention is used:
  - [Bracketed Description] – Description entries in variable tables surrounded with square brackets indicate the variable's placement in the *TmNS MIB*. For example: [tmnsTmaCommonIdentification 2] indicates that the variable is the second variable on the tmnsTmaCommonIdentification branch.

---

[4] Internet Engineering Task Force. "Structure of Management Information Version 2 (SMIv2)." April 1999. May be superseded or amended by update. Retrieved 18 April 2017. Available at https://datatracker.ietf.org/doc/rfc2578/.

[5] Internet Engineering Task Force. "Textual Conventions for SMIv2." RFC 2579. April 1999. May be superseded or amended by update. Retrieved 18 April 2017. Available at https://datatracker.ietf.org/doc/rfc2579/.

- Conventions used in place of table values include:
    - Blank String (“”) – A blank or empty string is indicated as double-quotes with no characters. This is commonly used to initialize a string before a value is assigned.
    - N/A – Not Applicable. For example, this value is given for the default state of tables indicating that the table has no rows, and so has no default values. N/A is also given for read-only variables that are expected to hold constant properties of the device (such as the *TmNSManageableApplication* type).

A.4. XML Concepts and Style Guide

A.4.a. Standards Language

The Metadata Standard defines a language. When compared to the other standards, the *Metadata* concept is closest to the MIB in the System Management standard. Both define a standard vocabulary for exchanging information. The MIB variables are somewhat like individual attributes and elements in a schema. A full language differs from a vocabulary in that in addition to identifying words and meanings, it also defines how the words can be composed together to form more-complex sentences. These concepts together are syntax, which identifies the words and valid sentence structures for a language. The semantics of a language are not merely related to the structure of the sentences, but also construct the meanings of the sentences in the context of the way the language will be used.

The Metadata Standard defines a language; the syntax identifies vocabulary and sentence structure, and the semantics provide meaning. The constraints in the Metadata Standard are distributed between statements that are syntax-related (encoded and enforced by the schema) and statements that are semantic-related (additional rules that are levied and provide meaning). The syntax of the language will be enforced using XML Schema constraints. When possible, XML mechanisms are used to enforce semantic constraints. In cases not supported cleanly by XML, text has been added directly to this standard. In such cases, the text will typically include the keyword “shall”. The phrase “the value of the `Name` element of the `Measurement` element shall be unique” is one such example.

Metadata instances (i.e., sentences) in general describe a telemetry system. The descriptions may be used in various ways: to configure *NetworkNodes*; to predict the performance of the network; or to capture requirements for the telemetry system.

A.4.b. General MDL Requirements

The MDL is an XML-based language for describing network-based telemetry systems. It can be used to capture requirements, design decisions, and configuration information. The MDL can also facilitate the interchange of information between tools.

This section provides context for how to interpret the language described herein, and suggests how it can be used. This includes:

- The drivers of the MDL design;
- The standards upon which it is built;
- How to extend and constrain the language.

A.4.c. XML Schema Concepts

The MDL defines a syntax, which includes a vocabulary, a set of types, and the rules for how an MDL instance document shall be structured. The syntax definition is realized using the XML Schema standard, which is maintained by the World Wide Web Consortium. This section explains the basic concepts of XML Schema that are utilized by the MDL. A more-detailed explanation of the fundamentals of the XML Schema standard is outside the scope of this document, but an explanation can be found in Section 2.2.2[6] of the W3C reference.

An XML Schema defines the rules of an XML-based language with six main constructs: elements, attributes, complex types, simple types, a root element, and constraints.

The XML Schema elements, of type `xsd:element`, represent information containers in an XML instance document. An element defines an XML tag that appears as "`<xsd:element>`" in an instance document. This specification defines where an element of the indicated type can be created in the instance document.

The XML Schema attributes, of type `xsd:attribute`, represent information that describe the element to which they are attached. The MDL has very few attributes defined because they are reserved for XML-specific uses. For example, they are used when the XML instance document needs to have information about the ordering of an element.

The XML Schema complex types, of type `xsd:complexType`, define structures that specify what an element can contain. Complex types are analogous to classes in an object-oriented language. An element defined as a complex type can contain other elements as well as attributes. They can define the combinations and ordering of the contained elements.

The XML Schema simple types, of type `xsd:simpleType`, define restrictions or specializations of basic types used within the schema definition. For instance, a simple type could be defined to restrict the value of an integer, of type `xsd:integer`, to an inclusive range of integer values from 0 to 255. These constructs are used mainly for validation and type restriction.

An XML Schema requires an instance document to have a top-level element called a root element. The root element contains all other elements and attributes in an instance document.

The XML Schema constraint mechanism defines a syntax (or grammar) of an XML language. Constraints enforce language rules against an XML instance document. For example, constraints can verify that referential integrity is maintained.

The XML Schema constraints can also be used to enforce semantic constraints in a very limited way. For example, constraints can be used to require an element to appear only if another element is defined; however, the XML Schema language does not have the ability to fully define the semantic context that is necessary to understand the full meaning of a language. An efficient and accepted approach for describing the semantics or meanings of a language has yet to be developed.

[6] World Wide Web Consortium. "Declaration Components" in *XML Schema* Part 1: Structures Second Edition. 28 October 2004. May be superseded by update. Retrieved 10 May 2017. Available at http://www.w3.org/TR/xmlschema-1/#Declarations_Summary.

A.4.d. Syntax Conventions of MDL Element Descriptions

Non-literal symbols (the ones that are not in “”) represent MDL elements or attributes. Each of these is linked to a section in this document.

By convention, this standard includes the built-in XML Schema types, which are identified with the namespace prefix “`xsd`”. For example, the `Name` element in the example above is of the type `xsd:string`. The supported simple types in the MDL are those defined in the XML Schema standard. Simple data types (i.e., `xsd:simpleType(s)`) defined by the MDL generally appear with the namespace prefix “`mdl`”.

A.4.e. Conditional Element in the MDL Schema Definition File

The MDL schema is a system-level description. Not all components require every element to properly configure. In such cases, these elements are conditional. The documentation specifies when the conditional elements shall be present and processed. Components not specifically called out in documentation of conditional elements shall not fail to configure if the particular conditional element is not present.

A.4.f. Naming Conventions in the MDL Schema Definition File

The Metadata Standard details the elements and attributes that form the MDL schema. In the MDL schema definition file, these MDL elements and attributes appear as instances of defined `xsd:complexType` and `xsd:simpleType` elements. Each declaration of these MDL-specific elements will specify a `name` attribute that is assigned a string that contains the name of the MDL element being described followed by a string suffix of “Type” or “Enum”. For example, the top-level element of the MDL schema is the `MDLRoot` element. In the MDL schema definition file, this element appears as an instance of an `xsd:complexType` element with a `name` attribute of “MDLRootType”. These `name` attribute strings that correspond to the defined MDL elements do not appear in this document; they only appear in the MDL schema definition file.

A.4.g. Indexing Policies

Numerical indices present in an MDL instance document are recommended to count starting at 1 and to increment by one (i.e., 1, 2, 3, 4,…).

A.4.h. Use of Supplemental XML-Based Standards

The use of other XML-based standards (i.e., other schemas) in conjunction with the MDL schema is permitted. Potentially, the use of these external standards through their accompanying schemas leverages existing knowledge to describe concepts and domains beyond those in the MDL. The MDL does not explicitly constrain the available mechanisms to use external standards; however, the linking of external schemas to the MDL schema shall not result in the modification of the MDL schema.

Refer to Chapter 23 Appendix 23-A for example approaches and mechanisms for linking other XML-based schemas to the MDL schema.

A.4.i. Uniqueness of ID Attributes

Values of `ID` attributes of any element in an MDL instance document shall be unique. The `ID` attributes are used to implement references.

A.4.j. Description of ReadOnly Element

All elements of type `xsd:complexType` in the MDL schema contain an optional `ReadOnly` element, which, of type `xsd:boolean`, indicates whether or not its containing element and all its subelements can be modified. A value of “true” indicates that these elements cannot be modified. Conversely, a value of “false” indicates that these elements can be modified. The default value of the `ReadOnly` element is “false”.

A.4.k. Description of Owner Element

All elements of type `xsd:complexType` in the MDL schema contain an optional `Owner` element, which can occur, at most, once in its containing element. The `Owner` element, of type `xsd:string`, is an identifier for the owner or administrator of the containing element in an MDL instance document. The rights and access controls associated with the identified owner will determine the ability of MDL instance document editors to modify the containing element and all its subelements.

NOTE	It is expected that a standardized set of values for the `Owner` element will be established. Until these values are determined, the Metadata Standard does not constrain the value of the `Owner` element.

APPENDIX 21-B

Bit Numbering and Byte Ordering

B.1. Bit Field Syntax

Numeric values specified in bit fields shall be represented using the following syntax:

```
size ' radix value
```

where

`size`	The number of binary bits that comprise the number.
`'`	A single quote separator.
`radix`	Radix of the number. Valid radix are: b = binary h = hexadecimal d = decimal
`value`	Bit field value represented as a numeric string.

Examples:

3'b101
32'h12345678
20'h1C (20'h0001C)
11'd123 (11'b00001111011)

NOTE	This bit field syntax is a subset of the Verilog Hardware Description Language syntax for representing numbers.

B.2. Bit Numbering Convention

Whenever an octet field represents a numeric quantity, the left-most bit in the field is the most significant bit (msb) and the right-most bit in the field is the least significant bit (lsb). Whenever a multi-octet field represents a numeric quantity, the left-most bit of the entire field is the msb.

When specific bits of fields are numbered, the msb is assigned the highest number, unless otherwise noted. For example, a 32-bit field is numbered from bit 31 down to bit 0, where bit 31 is the msb.

This bit numbering convention differs from the conventions defined in Chapter 4 and the IP specification. Table B-1 shows the differences between these different bit-numbering conventions.

Table B-1. Bit Numbering Conventions

Standard	Bit Numbering Convention	Single Octet Bit Numbering							
		msb							lsb
IRIG Chapter 21-28	lsb 0	7	6	5	4	3	2	1	0

IRIG Chapter 4	msb 1	1	2	3	4	5	6	7	8
IP Specification	msb 0	0	1	2	3	4	5	6	7
Example Octet Data (0xAB)		1	0	1	0	1	0	1	1

B.3. Octet (Byte) Ordering

Octet ordering is important for correct interpretation of multi-octet fields in all TmNS-specific message structures. Unless otherwise noted, these chapters specify the big-endian convention for octet ordering, which states that whenever a multi-octet field represents a numeric quantity, the most significant octet is transmitted first and stored in the memory location with the lowest address.

NOTE

The following table illustrates both big-endian and little-endian octet ordering for a 32-bit field with a value of 0x9A8B7C6D.

Big-endian Transmission Order/ Byte Address	0	1	2	3
32-bit field (4 bytes)	0x9A	0x8B	0x7C	0x6D
Little-endian Transmission Order/ Byte Address	3	2	1	0

APPENDIX 21-C

Citations

Institute of Electrical and Electronics Engineers. *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE 1588-2008. Geneva: International Electrotechnical Commission, 2008.

Internet Engineering Task Force. "Key Words for Use in RFCs to Indicate Requirement Level." RFC 2119. March 1997. May be superseded or amended by update. Retrieved 18 April 2017. Available at https://datatracker.ietf.org/doc/rfc2119/.

———. "Structure of Management Information Version 2 (SMIv2)." RFC 2578 April 1999. May be superseded or amended by update. Retrieved 18 April 2017. Available at https://datatracker.ietf.org/doc/rfc2578/.

———. "Textual Conventions for SMIv2." RFC 2579. April 1999. May be superseded or amended by update. Retrieved 18 April 2017. Available at https://datatracker.ietf.org/doc/rfc2579/.

National Institute of Standards and Technology. "Specification for the Advanced Encryption Standard (AES)." FIPS PUB 197. 26 November, 2001. May be superseded by update. Available at http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

World Wide Web Consortium. "Declaration Components" in *XML Schema* Part 1: Structures Second Edition. 28 October 2004. May be superseded by update. Retrieved 10 May 2017. Available at http://www.w3.org/TR/xmlschema-1/#Declarations_Summary.

*** * * END OF CHAPTER 21 * * ***